CS2105 Cheatsheet

github.com/reidenong/cheatsheets, AY24/25 Sem 1

Introduction to Networks

Data transmission

Circuit switching

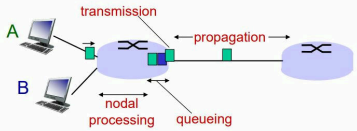
- Dedicated communication path established between two devices
- Resources reserved for duration of communication

Packet switching

- Data broken into packets of length L bits
- Packets transmited over the link at rate R
- Store and forward: Entire packet must arrive on a router before next transmission step

Delay, Loss and Throughput

Four sources of packet delay



d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

d_{queue}: queuing delay
- time waiting in the queue for transmission
- depends on congestion level of router

d_{trans}: transmission delay
- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay
- d: length of physical link
- s: propagation speed in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

Bit time: Transmission delay of 1 bit, ie. $\frac{1}{R}$
Packet loss: Packet arrives at router, but buffer is full
Throughput: How many bits can be transmitted per unit time
Link Capacity: Maximum theoretical throughput of a link

Protocol Layers

- **Application**: supports network applications (FTP, SMTP, HTTP)
- **Transport**: process-process data transfer (TCP, UDP)
- **Network**: routing of datagrams from source to destination (IP, routing protocols)
- **Link**: data transfer between neighbouring network elements (Ethernet, 802.11)
- **Physical**: bits on the wire

Application Layer

HTTP

Non-persistent HTTP

- At most one object sent over a TCP connection, connection closed

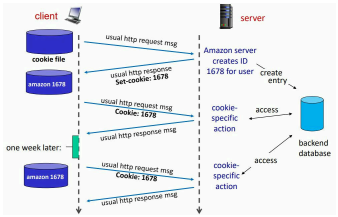
Persistent HTTP

- Multiple objects sent over single TCP connection

HTTP/1.0	HTTP/1.1
Default non-persistent unless 'keep-alive' header is used	Default persistent
HEAD: Server leaves object out of response	PUT: Upload file to url path DELETE: Delete file at url path

Cookies

- HTTP is stateless: Server maintains no information about past client requests. Server may maintain state using cookies



Conditional GET

- Don't send object if client has up-to-date cached copy

DNS

DNS translates between Hostname and IP address with right to left resolution.

- Once a name server learns mapping, it caches it until time-to-live (TTL) expires
- Runs over UDP
- Iterative query: Local DNS server queries all other servers
- Recursive query: Local DNS server queries root, root queries top-level domain, etc.
- Hierarchy: Local -> Root -> Top-level domain -> Authoritative

Sockets

Socket is the software interface between app processes and transport layer protocols.

TCP sockets
- Connection-oriented, reliable, in-order byte stream
- When contacted by client, server TCP creates new socket
- Server uses client IP and port to distinguish clients
- Client creates socket, establishes TCP connection to server

UDP sockets
- Connectionless, unreliable, unordered datagrams
- Server uses one socket to serve all clients
- No connection setup
- Sender attaches destination IP and port to each datagram
- Transmitted datagram may be lost or out of order

Transport Layer

UDP

- Multiplexing at sender: UDP gathers data from processes, adds UDP header, passes to network layer
- Demultiplexing at receiver: UDP extracts data, passes to correct process
- When UDP receiver receives a segment, it checks the destination port in segment. Datagrams from different sources with the same port will go to the same socket at destination

UDP header

UDP header is relatively small, with 4 parts of 16 bits each.

- source port #: port of sending process
- dest port #: port of receiving process
- length: length of UDP header + data
- checksum: error detection

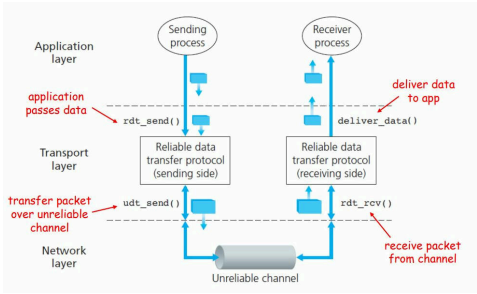
UDP checksum

Detects errors in transmitted segment

- Sender: Treats segment as sequence of 16-bit integers, checksum is sum of all integers
- Receiver: Adds all 16-bit integers, if result is 1's complement of sum, no error

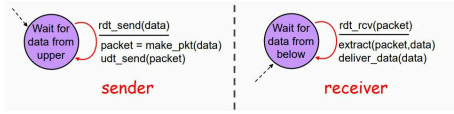
TCP (Transmission Control Protocol)

Principles of Reliable Data transfer



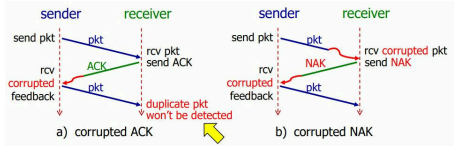
reliable data transfer (rdt) protocols

rdt 1.0: Perfectly reliable channel



rdt 2.0: Channel with bit errors

- Channel may flip bits in packets
- Receiver may use checksum to detect bit errors. They may signal to sender ACK or NAK
- Sender retransmits on NAK, or corrupted ACK/NAK
- May cause transmission of correctly received packet



rdt 2.1: Channel with bit errors

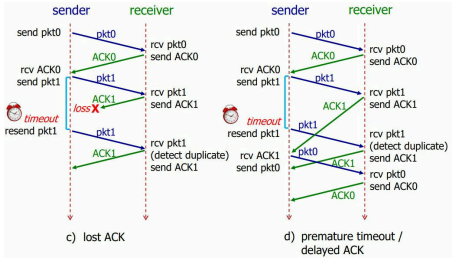
- Same as rdt 2.0, but Sender adds sequence number to packet
- Sender still retransmits on NAK, or corrupted ACK/NAK
- Receiver only accepts packet with correct sequence number

rdt 2.2: NAK-free

- Receiver sends ACK for last received packet
- Sender retransmits packet x until ACK(x) is received

rdt 3.0: Channel with bit errors and loss

- Sender retransmits after timeout if no ACK is received



Go-back-N pipelining protocol

- underlying channel may flip bits in packets, lose packets, incur delay, but will not reorder
- **GBN Sender**
  - up to N unACKed packets in pipeline
  - uses sliding window starting at lowest unACKed packet
  - on receiving ACK(x), move window to $x+1$ and send all unsent packets in window
  - maintains timer for oldest unACKed packet. on timeout, retransmit all packets in window
  - ignores duplicate ACKs
- **GBN Receiver**
  - only accepts in-order packets (ie. only remembers expected sequence number)
  - sends cumulative ACK: ACK m indicates all packets up to m have been received

Selective Repeat pipelining protocol

- **SR Sender**
  - maintains timer for each unACKed packet, retransmits only when timer expires
  - maintains sliding window starting at lowest unACKed packet
  - on receiving ACK(x), move window to lowest unACKed packet
- **SR Receiver**
  - sends individual ACKs for each correctly received packet
  - buffers out-of-order packets until missing packet is received

Performance Analysis

	Throughput	Utilization
Stop&wait	$\frac{L}{\text{RTT} + d_{\text{trans}}}$	$\frac{d_{\text{trans}}}{\text{RTT} + d_{\text{trans}}}$
Pipelining, window x	$\frac{x \cdot L}{\text{RTT} + d_{\text{trans}}}$	$\frac{x \cdot d_{\text{trans}}}{\text{RTT} + d_{\text{trans}}}$

Utilization is the fraction of time the channel is transmitting data.

TCP Specifications

- Point-to-point: one sender, one receiver
- Connection-oriented: handshaking before data exchange
- Full duplex: data flows in both directions
- Reliable, in-order byte stream

Connection-oriented De-multiplexing

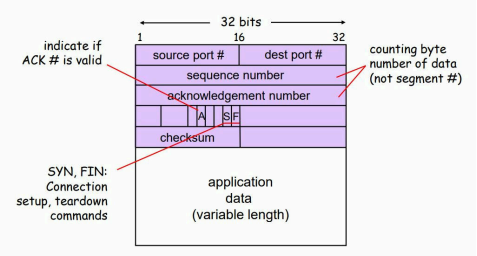
A TCP socket is identified by a 4-tuple:
`(sourceIP, sourcePort, destIP, destPort)`

- A host server can have multiple connection sockets with the same IP and port connected to different client IPs and ports
- The server uses the 4-tuple to identify the correct socket to receive and send data

TCP Buffers, segments

- Two buffers are created after handshaking at any side of the connection: send buffer and receive buffer
- A TCP segment has a Maximum Segment Size (MSS) of 1460 bytes, excluding the header

TCP Header



- `sequence_number` is the byte number of the first byte in segment
- `acknowledgement_number` is the next byte expected by the receiver
- `A` is the ACK flag signifying ack number is valid
- `S` is the SYN flag for connection establishment during handshake
- `F` is the FIN flag for connection termination

TCP Sender Events

Application passes data to TCP

1. create TCP segment with sequence number `NextSeqNum`
2. start timer if not already running
   - only one timer for oldest unACKed packet
3. pass segment to IP
4. `NextSeqNum += length(data)`

Timer timeout

1. retransmit unACKed segment with smallest sequence number
   - Retransmit only oldest unACKed packet
2. restart timer

ACK(x) received

1. if `x > SendBase`, set `SendBase = x`
   - ACK is cumulative
2. start timer if there are still unACKed segments

TCP ACK Generation

TCP ACK is the expected in-order seq number and is cumulative.

Event at TCP receiver	*TCP receiver action*	
Arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed	Delayed ACK: wait up to 500ms for next segment. If no next segment, send ACK	
Arrival of in-order segment with expected seq #. One other segment has ACK pending	Immediately send single cumulative ACK, ACKing both in-order segments	
Arrival of out-of-order segment higher-than-expect seq. # (gap detected)	Immediately send *duplicate ACK*, indicating seq. # of next expected byte	
Arrival of segment that partially or completely fills gap	Immediately send ACK, provided that segment starts at lower end of gap	

TCP Timeout Value

- Timeout too short: premature timeout, unnecessary retransmissions
- Timeout too long: slow reaction to segment loss
- Timeout must be longer than RTT, but RTT varies

- TCP uses exponential weighted moving average of RTT with a bias towards the most recent RTT

$$\text{Est\_RTT}_{N+1} = (1-\alpha) \cdot \text{Est\_RTT}_N + \alpha \cdot \text{Sample\_RTT}$$

$$\text{Dev\_RTT}_{N+1} = (1-\beta) \cdot \text{Dev\_RTT}_N + \beta \cdot |\ \text{Sample\_RTT} - \text{Est\_RTT}_{N+1}\ |$$

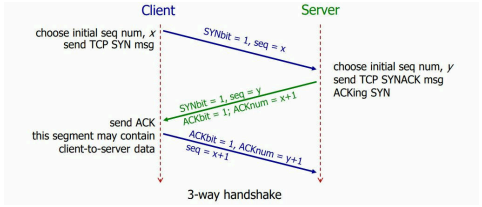
$$\text{Timeout} = \text{Est\_RTT} + 4 \cdot \text{Dev\_RTT}$$

TCP Fast Retransmission

- Timeout is often relatively long, resulting in a long delay before resending a lost packet.
- Fast retransmission: if sender receives 3 duplicate ACKs (4 total), it assumes that the segment was lost and retransmits it immediately

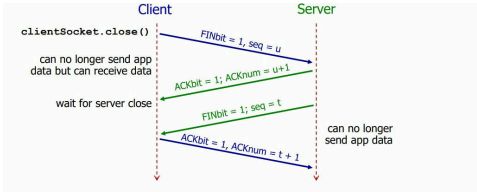
Establishing connection

3-way handshake



Terminating connection

4-way handwave



Network Layer

IP address

A 32-bit integer used to identify a host or router. A IP address is associated with a network interface.

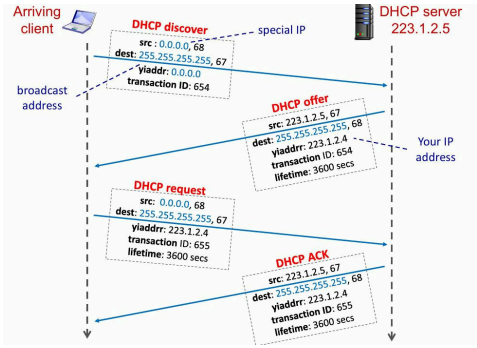
Dynamic Host Configuration Protocol

DHCP allows a host to dynamically optain IP address from server when it joins network.

4-step process

1. Host broadcasts `DHCP discover` message
2. DHCP server responds with `DHCP offer` message
3. Host requests IP address with `DHCP request` message
4. DHCP server sends `DHCP ack` message

DHCP ACK contains client IP address, IP for 1st-hop router, and IP of DNS server + name.



Special IP addresses

Special Addresses	Present Use
0.0.0.0/8	Non-routable meta-address for special use
127.0.0.0/8	Loopback address. A datagram sent to an address within this block loops back inside the host. This is ordinarily implemented using only 127.0.0.1/32.
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	Private addresses, can be used without any coordination with IANA or an Internet registry.
255.255.255.255/32	Broadcast address. All hosts on the same subnet receive a datagram with such a destination address.

Subnets

A subnet is a network formed by a group of directly interconnected hosts. These hosts can physically reach each other without intervening routers.

The internet's IP address assignment strategy is Classless Inter-Domain Routing. It allows for more efficient use of IP addresses by allowing for variable-length subnet masks.

Routing

The internet consists of a hierarchy of Autonomous Systems (AS) each consisting of it's own routers and links. Intra-AS routing is handled by routers within the AS, while Inter-AS routing is handled by routers between ASes.

Intra AS routing

Routing within an AS is finding a between a source and destination within the same AS. The routing algorithm is usually OSPF or RIP.

Routing Algorithms classification

"Link State" algorithms are when all routers have complete topology information. They then use dijkstra to compute the shortest path locally.

"Distance Vector" algorithms are when routers only know the distance to their neighbours. They then exchange information with their neighbours to compute the shortest path.

RIP

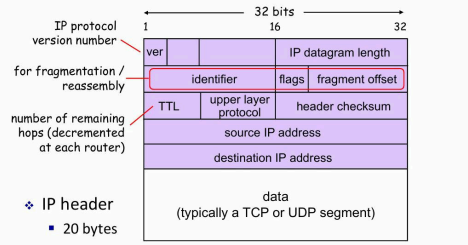
RIP (Routing Information Protocol) implements *dv* routing. It uses hop count as the cost metric, and exchanges routing table every 30s over port 520. If a router does not receive an update from a neighbour for 3 minutes, it is marked as unreachable.

NAT

NAT (Network Address Translation) allows a router to translate private IP addresses to public IP. One connection has one entry in the NAT table.

1. Host (user) sends datagram to destination through router.
2. Router changes datagram source address and port, and updates NAT translation table.
3. Router forwards datagram to destination.
4. Reply arrives to router.
5. Router changes datagram destination address and port.

IPv4 Datagram



IP Fragmentation

Links between routers have different MTU (Maximum Transfer Units), which is the maximum amount of data a link-level frame can carry. If a datagram is larger than the MTU, it is fragmented into smaller datagrams.

The content of a IP datagram may be broken up into smaller fragments, each with a new IP header. The fragments are reassembled at the destination.

- Flag (frag flag) indicates if there is a next fragment from the same segment.
- Offset indicates the position of the fragment in the original datagram divided by 8. (ie. the second fragment of a 480 byte datagram has an offset of 60)

ICMP

ICMP (Internet Control Message Protocol) is used by routers and hosts to communicate network-level information. It is used for error reporting and debugging. *ping* sees if a remote host is reachable, *traceroute* shows the path taken to a destination.

Link Layer

The link layer sends datagrams between adjacent nodes over a single link. IP datagrams are encapsulated into link-layer frames for transmission.

As we wish to send data between N nodes via cable, there is a need to manage link addressing, link protocol and error detection.

Types of network links

- Point-to-point: single wire, 2 hosts
- Broadcast: shared medium, multiple hosts

Multiple Access protocols

- Channel partitioning: divide channel into smaller pieces (time, frequency, slots) and allocate a piece to each node for exclusive use
- Taking turns
- Random access: nodes transmit whenever they have data to send. Collisions are possible, and must be resolved.

Ideal multiple access protocol

Given a broadcast channel of R bps, we want

1. **Collision free**
2. **Efficient**: When only 1 node is sending, it has rate R
3. **Fairness**: when M nodes want to transmit, each can send at rate $\frac{R}{M}$
4. Fully **decentralized**: no special node to coordinate transmissions

Random Access Protocols

Slotted ALOHA

We have multiple nodes sending data to a central receiver. All frames are of equal length L, and time is divided into slots of length $\frac{L}{R}$. Nodes can only send at the start of a slot. If a collision occurs, the nodes will retransmit in the next slot with probability p.

Collision Free	No
Efficiency	For one node, throughput is R. When there are many active nodes, maximum effiency is 1/e as slots are wasted due to collisions / empty.
Fairness	Yes
Decentralized	Yes

Pure ALOHA

Whenever a node has a fresh frame to send, we transmit immediately. If a collision occurs, we wait for 1 frame tranmission time and retransmit with probability p.

Pure ALOHA has a higher collision rate, as a frame sent at t_0 can collide with any frame send within $(t_0 - 1, t_0 + 1)$. Same properties as Slotted ALOHA, but lower efficiency at around 0.18.

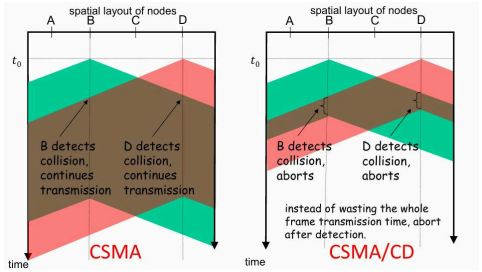
CSMA

A major flaw in ALOHA is that a node transmits regardless of whether the channel is busy. Carrier Sense Multiple Access (CSMA) waits for the channel to be idle before transmitting.

Propagation delay can stil exist where each node may not hear the transmission of another node immediately, and this can lead to collisions.

CSMA/CD

CSMA/CD (Collision Detection) allows a node to abort transmission if a collision is detected. The node then waits a random amount of time before retransmitting.



However, the probability of collision in subsequent time slots remains the same, or may worsen if a new node begins to transmit. The Backoff Algorithm aims to adapt retransmission based on estimated current load, and more collisions → implied heavier load → longer backoff interval.

For Binary exponential backoff, backoff interval doubles each collision. For the mth collision, we choose k from $\mathbb{Z} \cap [0, 2^m - 1]$ and wait for k slots.

A frame size that is too small results in collisions going undetected. This is avoided if $\max(d_{\text{prop}}) \leq d_{\text{trans}}$, where d_{prop} is the propagation delay and is proportional to the network diameter, and d_{trans} is the transmission delay, directly proportional to the frame size. CSMA and variants impose restrictions on minimum frame size to ensure collisions will always be detected.

For Ethernet, 1 time unit is set as 512 bit transmission times.

Collision Free	No
Efficiency	Yes
Fairness	Yes
Decentralized	Yes

Channel Partitioning Protocols

In Time Division Multiple Access (TDMA), each node gets a fixed length slot in a round-robin fashion.

In Frequency DMA, the channel is divided into frequency bands, and each node is assigned a band.

Collision Free	Yes
Efficiency	Poor. For one node, throughput is capped at $\frac{R}{N}$ for N nodes. Unused slots are also wasted.
Fairness	Yes
Decentralized	Yes

Taking Turns Protocols

Master Node

A master node polls each of the nodes in a round-robin fashion.

Collision Free	Yes
Efficiency	Higher than channel partitioning. Only has polling overhead.
Fairness	Yes
Decentralized	No, relies on master node which a singular point of failure

Token Passing

A token is passed around a cyclical network, and the node with the token can transmit a maximum amount of frames and then passes the token to the next node.

Collision Free	Yes
Efficiency	Yes, but has overhead of token passing
Fairness	Yes
Decentralized	Yes
Downside	Node failure can break the ring, and a token loss can be disruptive

Error Detection

1D Parity checking

WLOG, in an even parity scheme we include an extra bit such that the total number of 1s in the data (including parity bits) is even. This allows us to detect all odd number of single bit errors. However, since errors are often clustered together, the probability of undetected errors in a frame can approach 50%.

2D Parity checking

We divide the data into i rows and j columns, and add parity bits for each row and column. This allows us to detect and correct single bit errors in the data, and detect two bit errors and any odd number of single bit errors.

Cyclic Redundancy checks

We wish to send a binary message of D and R CRC bits. Both sender and receiver have a generator number G of $R + 1$ bits.

1. Append R 0s to the message D to get D'
2. Divide D' by G to get a remainder of R bits, which are our CRC bits.

The receiver divides the received message by G, and if the remainder is 0, the message is accepted. Otherwise, the message is rejected.

CRC can detect all odd number of single bit errors, all burst errors of up to R bits, and all burst errors of $\geq R$ bits with a probability of $1 - 2^{-R}$.

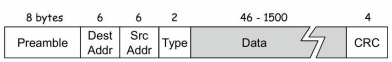
Local Area Networks

LAN is a computer network that interconnects computers within a geographical area, and consists of technologies such as the IBM Token Ring, Ethernet, Wi-Fi, and Async Transfer Mode (ATM).

802.3 Ethernet

Frame structure

A Network Interface Card (NIC) encapsulates an IP datagram into an Ethernet frame.



If NIC receives a frame with matching destination or broadcast MAC address, it passes the data to the network layer protocol, otherwise it discards the frame. The maximum size of the data is 1500 bytes, and the minimum size is 46 bytes to ensure collision detection. Corrupted frames are dropped. The `Type` references higher level protocols, since hosts can use network-layer protocols besides IP, and allows Ethernet to multiplex network layer protocols.

The `Preamble` consists of

- 7 bytes of `1010 1010`
- 1 byte of `1010 1011`

This helps the receiver to synchronise with the sender's clock. It provides a square wave pattern that informs the receiver of the sender's clock rate, which is important when trying to distinguish between long sequences of the same bit.

Ethernet Data Delivery

- Unreliable as the receiving NIC do not use ACKs to confirm receipt, and data in dropped frames are recovered only if the sender uses a higher level protocol for rdt.
- Ethernet uses CSMA/CD with binary exponential backoff for multiple access.

Ethernet Topologies

- **Bus**: All nodes share a single communication line as a broadcast LAN where all nodes may collide with each other. If the backbone cable is damaged, the entire network fails. The presence of collisions also makes this very slow and not ideal for large networks.
- **Star with Hub**: All nodes are connected to a central hub, a physical layer device acting on individual bits rather than frames. It retransmits any received bit to all other interfaces. While cheap and easy to maintain, the presence of collisions makes this slow and bad for large networks.
- **Star with Switch**: All nodes are connected to a central switch, a layer-2 device that works on frames rather than individual bits. It has no collisions, and works by store and forwarding.

Ethernet Switch

A switch is a link-layer device that receives incoming frames, learns its MAC address, and then selectively forwards the frame to one or more outgoing links. It stores and forwards ethernet frames, and uses CSMA/CD to access links. The switch is transparent to the host, and switches do not require any configuration.

Each node has a dedicated, direct connection to the switch, and switches buffer packets between each interface. This enables no collisions. By reading the MAC address of incoming frames, the switch learns which hosts can be reached through which interfaces, and records this in the switch table in the form < `MAC addr` , `interface` , `TTL` >. The switch forwards the frame to the correct interface, and if the destination is not in the table, it floods the frame to all interfaces except the incoming one.

When a frame is received at switch

1. Record source MAC address and interface
2. Index switch table
3. If destination MAC address from the segment it came from, filter (drop frame)
4. If destination MAC address is in the table, forward to the correct interface
5. If destination MAC address is not in the table, flood to all interfaces except the incoming one

Switches vs Routers

- Switches operate at the link layer and check MAC addresses, while routers operate at the network layer and check IP addresses.
- Routers compute routes to destination, whereas switches just forward frames.
- Both use store and forward.

MAC addresses

A MAC address is a 48-bit address assigned to a NIC. Upon a NIC receiving a frame, if the dest matches, it passes the data to the network layer protocol, else it discards the frame.

Where IP addresses are used to move IP datagrams from source to destination, MAC addresses are used to move frames over every single link. IP addresses are also hierarchical, while MAC addresses are flat.

ARP

If we know the IP address of a host, we can use Address Resolution Protocol (ARP) to find the MAC address of the host.

Sending Within the same Subnet

Each IP node has an ARP table, which stores the mappings of the IP address and MAC address of other nodes in the same subnet.

- If B's MAC address is in A's ARP table, A sends the frame to B.
- If B's MAC address is not in A's ARP table, A broadcasts an ARP query packet with dest set to `FF-...-FF` and source set to A's MAC address. B receives the packet and sends an ARP reply packet with its MAC address to A. A can then cache B's IP and MAC address in its ARP table.

Sending to other Subnets

If the destination IP address is not in the same subnet, the sender sends the frame to the router. A creates a IP datagram with IP source A and dest B. This is then encapsulated in a link layer frame with the router's MAC address as the destination.

At the router, the router decapsulates the frame and passes it to IP. the router then forwards the same datagram in a link-layer frame with the dest MAC address of B.

Network Security

Terminologies

- **Confidentiality**: Only authorised parties can understand data
- **Integrity**: Ensure data is not altered
- **Authentication**: Ensure the identity of the sender
- **Eavesdropping**: intercept messages
- **Impersonation**: Spoofing fields in packet
- **Hijacking**: Replacing sender or receiver
- **Denial of Service**: Preventing use of service by others.

Cryptography

Ciphers

- Caesar Cipher shifts each letter by a fixed number of positions. Key = Shift value, 25 different keys.
- Monoalphabetic ciphers substitue one letter for another, and has 26! keys of possible mappings. These are easily broken by statistical analysis methods ie. considering frequency.
- Polyalphabetic ciphers have a key of n substitution ciphers and a cycling pattern.
- Block ciphers break the message to be encrypted into block of K bits, and then encrypts each block using a 1-1 mapping. It has $2^K!$ keys.
  - DES (Data Encryption Standard) 56-bit sym key, 64 bit block
  - AES (Advanced Encryption Standard) 128, 192, 256 bit keys, 128 bit block

Symmetric Key Cryptography

Both sender and receiver have the same symmetric key K_s. Each key is unique to a pair of individuals, and the message can only be encrypted or decrypted by the sender and receiver.

Public Key Cryptography

The sender uses a public encryption key known to all, while the receiver uses a private decryption key. It has the following requirements:

1. We need public key $K_B^+(.)$ and private key $K_B^-(.)$ s.t. $K_B^-(K_B^+(m)) = m$
2. Impossible to determine private K_B^- from public K_B^+

RSA

Finding public/private key pair

1. Choose two large prime numbers p and q
2. Compute $n = pq$, $z = (p-1)(q-1)$.
3. Choose e s.t. $e < n$ and $\gcd(e, z) = 1$
4. Choose d s.t. $ed \equiv 1 \bmod z$
5. Public key K_B^+ is (n, e), private key K_B^- is (n, d)

Encryption/decryption

1. Encrypt m with K_B^+: $c = m^e \bmod n$
2. Decrypt c with K_B^-: $m = c^d \bmod n$

In general,

$$(m^e \bmod n)^d \bmod n = m = m^{ed} \bmod n = \left(m^d \bmod n\right)^e \bmod n$$

If we know $K_B^+ = (n, e)$, it is still difficult to find d, as we need to factorise n into p and q to find z and d, which is difficult when n is large.

Session keys

Exponentiation is computationally expensive, and DES is much faster than RSA but requires a shared key. A solution is to use RSA to exchange a session key, which is then used to encrypt the message with DES.

Message Integrity

Sender, receiver want to ensure messages are not altered without detection. Checksums are designed to detect accidental errors and not attacks, while CRC is not sufficient as output is biased to input, ie. minor changes can be made to the message without detection.

Cryptographic Hash Function

A function $H(.)$ that takes an input m and produces a fixed-size fingerprint, while making it computationally infeasible to find two different messages x and y such that $H(x) = H(y)$. md5sum is a hash function that generates short fixed-length digests of 128 bits.

Hashes differ from encryption as they are one-way functions and have deterministic outputs.

Message Authentication Code

Sender and receiver share authentication key s. To ensure integrity, we send $(m, H(m+s))$, so that third parties cannot forge the message without knowing s.

Authentication

Sender, receiver want to ensure the identity of the other party. We can achieve this with digital signatures, which are

1. Verifiable: Recipient can check if the signature and message were generated by the sender
2. Unforgeable: Only the sender can generate the signature and message
3. Non-repudiable: Recipient can convince others only S could send the message

This can be done with the property of RSA keys: Sender encrypts message with private key to create $(m, K_S^-(m))$. Receiver can then decrypt with public key $K_S^+(.)$ and check for equality with message. Equality means whoever signed m must have used the sender's private key.

Public Key Distribution

1. Public Announcement
2. Public Directory
3. Public Key Infrastructure (PKI)
   - Certificate Authority (CA) issues certificates binding public keys to entities.
   - Certificates include identity, public key, time window, signature of the CA
   - CAs issue and sign digital certificates to websites, and maintain a directory of public keys. People sending certified public keys first verify the CA's public key to be sure the public key is authentic.
   - Operating systems has a list of "Trusted Root Certification Authorities".

Access & Availability

Firewalls

Firewalls isolate an organization's internal net from the larger internet, allowing some packetrs to pass but blocking others. They do several things:

1. Prevent denial of service (DoS) attacks
   - SYN flooding occurs when attacker establishes many bogus TCP connections, exhausting resources.
2. Prevents illegal modification of internal data.
3. Allows only authorized access to internal network.

Firewalls come in 3 types:

- Stateless packet filters
- Stateful packet filters
- Application gateways

Stateless packet filtering

Router filters packet by packet, makes decision based on

- Source, Dest IP address
- TCP/UDP source and destination port numbers
- ICMP message type
- TCP SYN, ACK bits

Example of policies:

Policy	*Firewall Setting*
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution' s public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Limitations of Firewalls

- Cannot protect against IP Spoofing
- Can be a bottleneck, slow down packets
- Incurs a tradeoff between degree of communication with the outside world against level of security.

Secure Email

Sender wants to provide secrecy, authentication and integrity. Sender generates symmetric key K_s, encrypts message with K_s and K_s with recipient's public key. Sender may also digitally sign message with sender's private key. A total of 3 keys are used- sender's private key, recipient's public key, and symmetric key.